

Southern California Linux Expo 2024

# Secure Consumption of Open Source Software: Evaluating, Utilizing, and Contributing Safely

Katherine Druckman  
Open Source Security Evangelist

Why are we here?

Today we'll cover

 Security challenges in consuming open source software

 Evaluating open source projects through a security lens

 Project health, governance, management, and community

 Tools for securing open source software

 Open source community security efforts: OpenSSF

 How we can contribute to a safer ecosystem

Security challenges

# Why is open source security so challenging?

# Open source is *everywhere*

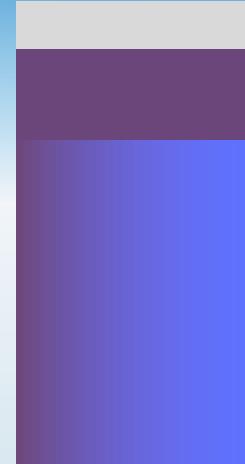
**96%**  
of codebases



Source: Synopsis

**77%**  
of code within

**70–90%**  
of all software



Source: Linux Foundation

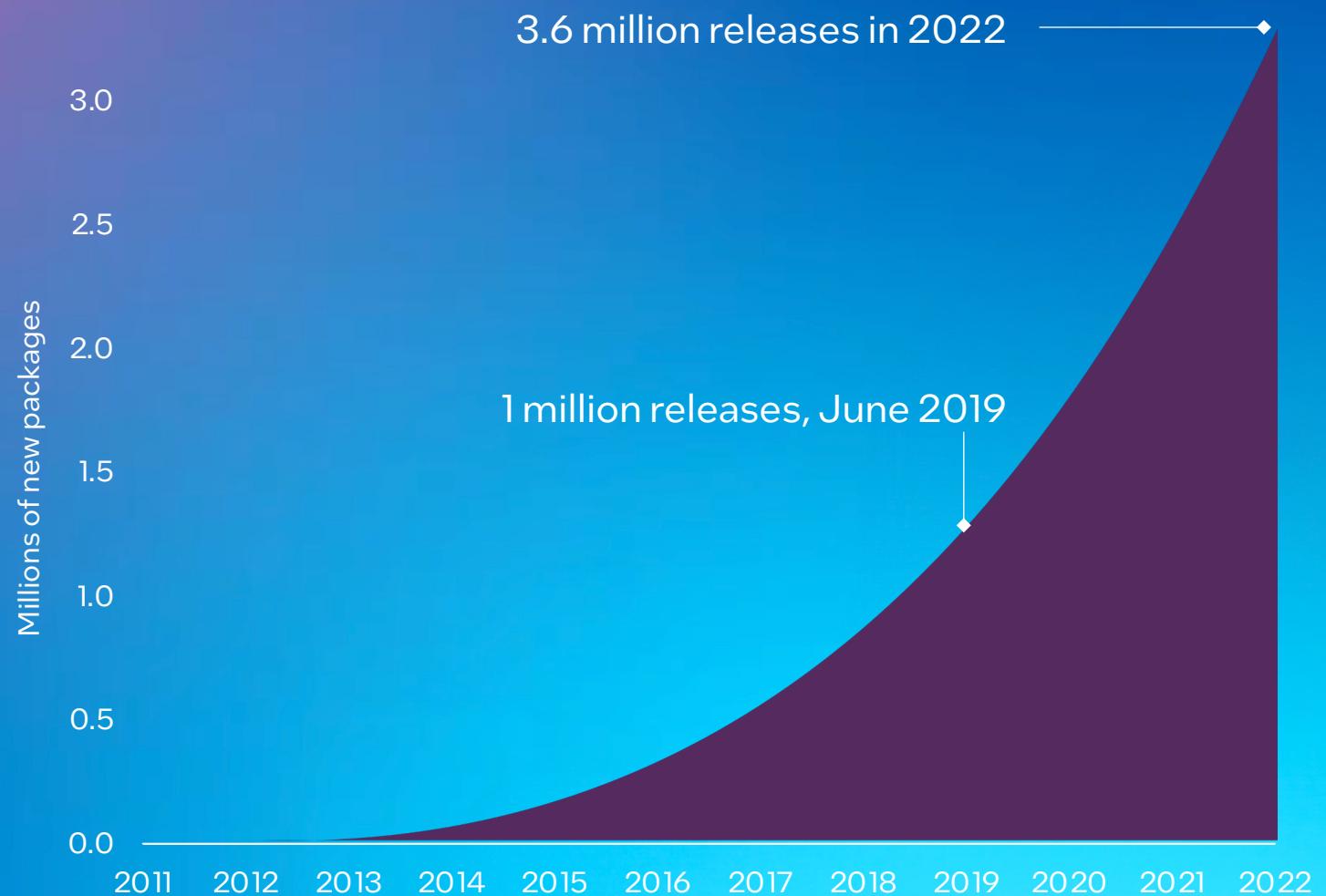
Open source is *everywhere*



Yay! We won!

## Node Package Manager (NPM) | New project releases over time

Millions of packages ...



Millions of packages ...  
dozens of maintainers

## Packages by number of maintainers



# It's not just NPM

[npmjs.org](https://npmjs.org)

3,639,251 packages  
41,815,290 versions  
822,231 maintainers  
222,295 namespaces  
742,169 keywords  
237,045,471,901 downloads



[proxy.golang.org](https://proxy.golang.org)

1,105,378 packages  
9,970,233 versions  
449,953 namespaces  
70,056 keywords



[hub.docker.com](https://hub.docker.com)

1,001,771 packages  
10,844,967 versions  
411,451 namespaces  
1,713 keywords  
334,237,037,105 downloads



[nuget.org](https://nuget.org)

624,254 packages  
7,566,455 versions  
85,940 maintainers  
129,237 keywords  
514,134,790,590 downloads



[pypi.org](https://pypi.org)

542,396 packages  
5,603,074 versions  
229,691 maintainers  
193,748 keywords  
29,768,771,520 downloads



[repo1.maven.org](https://repo1.maven.org)

499,556 packages  
11,361,001 versions  
66,310 namespaces  
31,287 keywords



## Statistics

Registries: 59

Packages: 8,840,726

Versions: 97,334,410

Namespaces: 1,328,127

Maintainers: 1,440,484

Downloads: 1,811,443,044,372

Keywords: 1,521,288

# Common Vulnerabilities and Exposures (CVE)

CVE is a dictionary of common names for publicly known cybersecurity vulnerabilities, each of which receives a CVE Identifier.

CVE Identifiers make it easy to share data across separate network security databases and tools. Plus, they provide a baseline for evaluating the coverage of an organization's security tools.

- CVE = Common Vulnerabilities and Exposures
- List of “all” publicly known software security vulnerabilities starting in 1999
- MITRE Corporation manages and maintains CVE on behalf of the US National Cybersecurity Division



# CVEs: True or false?

A project with no CVEs is more secure than a project with many CVEs.

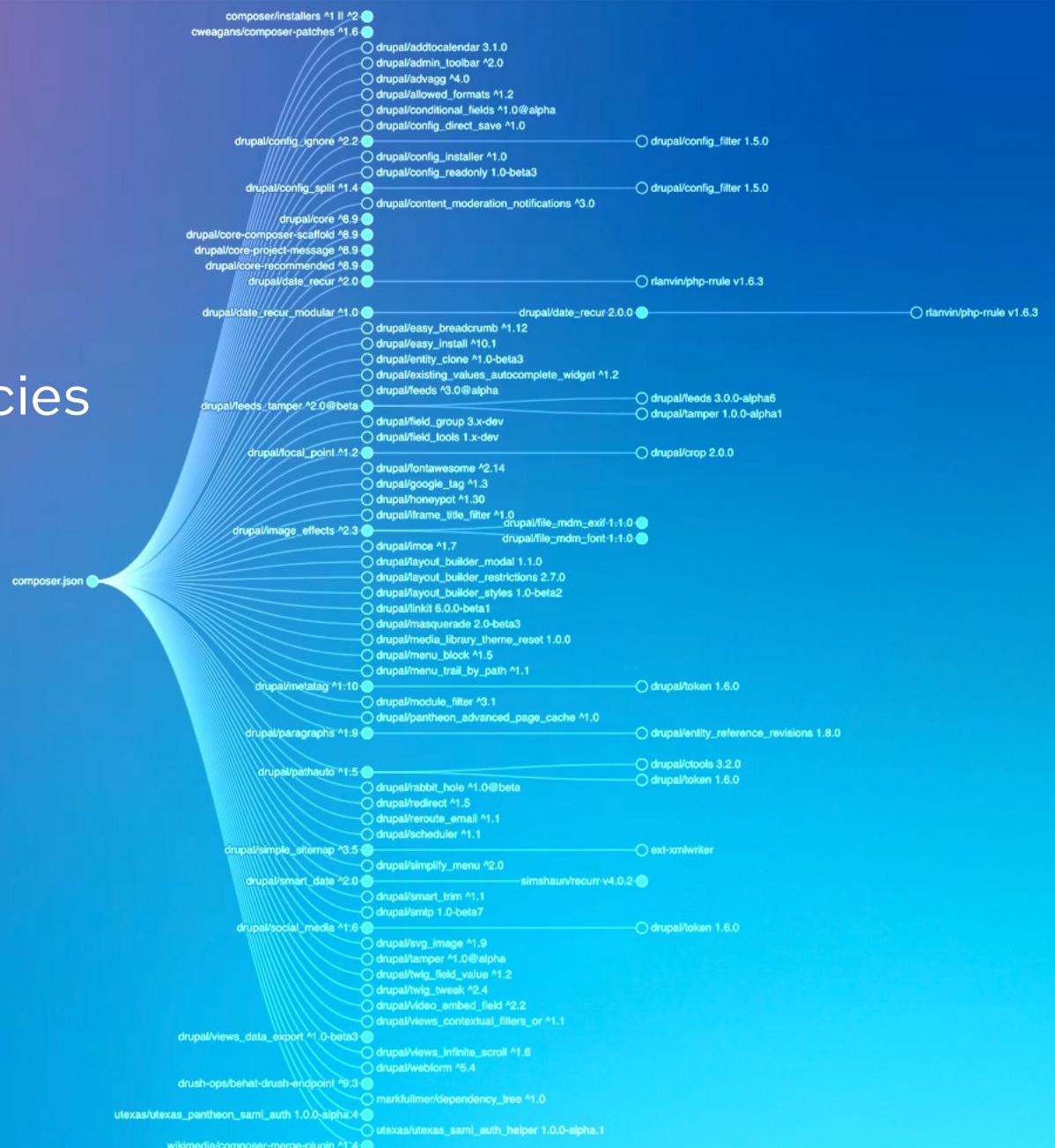


# CVEs per year



# So many dependencies

- Secondary and tertiary dependencies can get well into the 100s...
- Especially with web applications



# New contributors!

From GitHub:

- 2023 had the largest number of first-time contributors
- 420 million total projects (27% growth YOY)
- 4.5 billion contributions in 2023



Evaluating projects

# What does it mean to consume open source software securely?

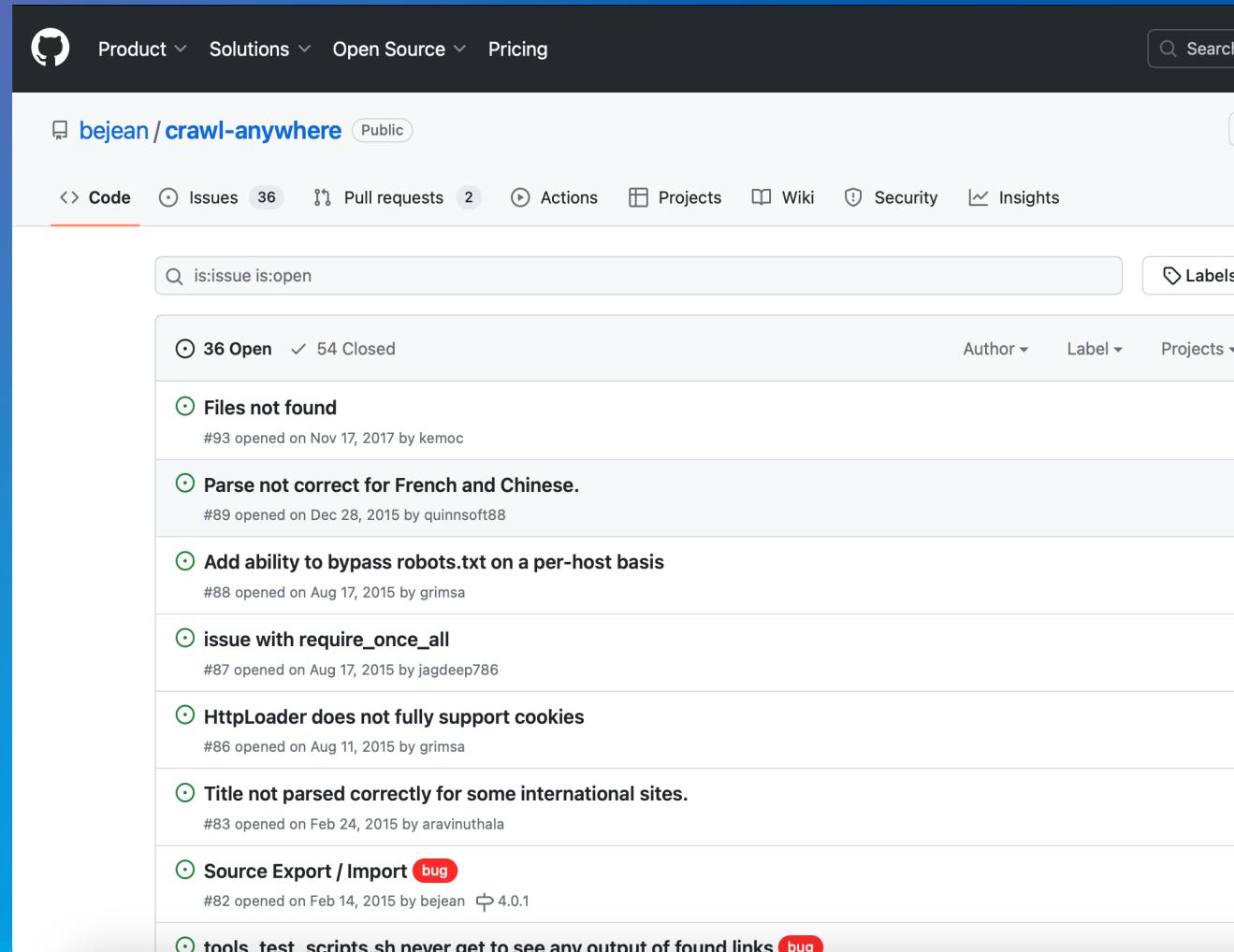
# Evaluating open source projects

1. Review basic health—Is it active?
2. Check governance—Is it defined?
3. Review maintenance & releases—Is there a cadence?
4. Explore the community—Are people engaged?
5. Bug reporting—Is there a documented process?

What's the first thing you would look at when evaluating an open source project to use or include as a dependency?

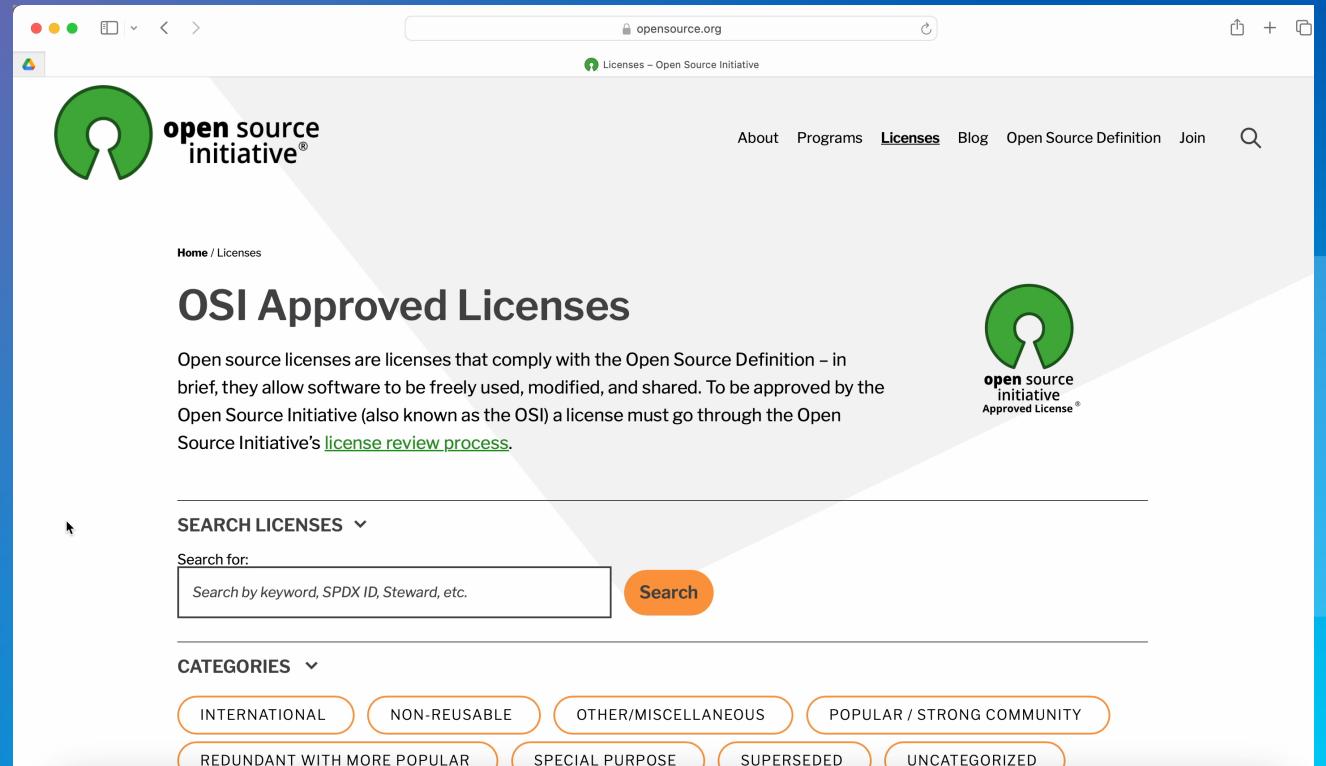
# 1. Evaluating software: Basic health

- Does the project even have a maintainer anymore?
- When was the last commit?
- Look at the issue queue
  - How active is it?
  - When was the last post?
  - When was the last response to an issue?



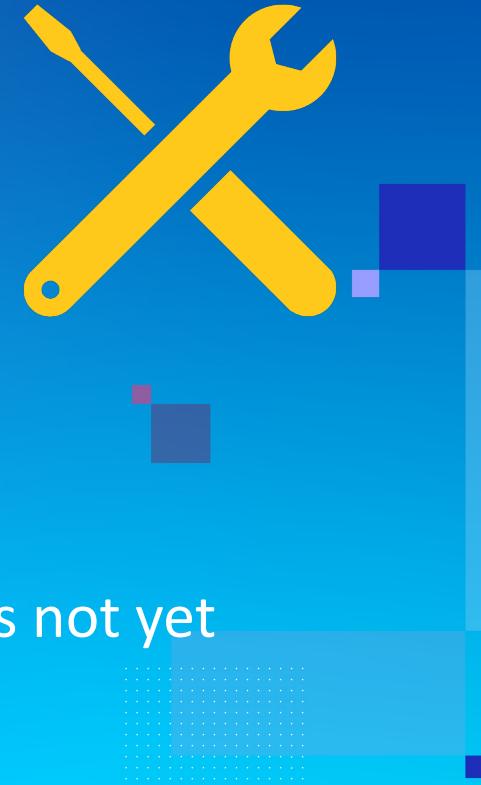
## 2. Evaluating software: Governance

- Clearly defined governance?
  - Clearly stated license? (Hopefully OSI approved)
  - More than one maintainer
  - Maintainers from more than one company or organization
  - How are decisions made?



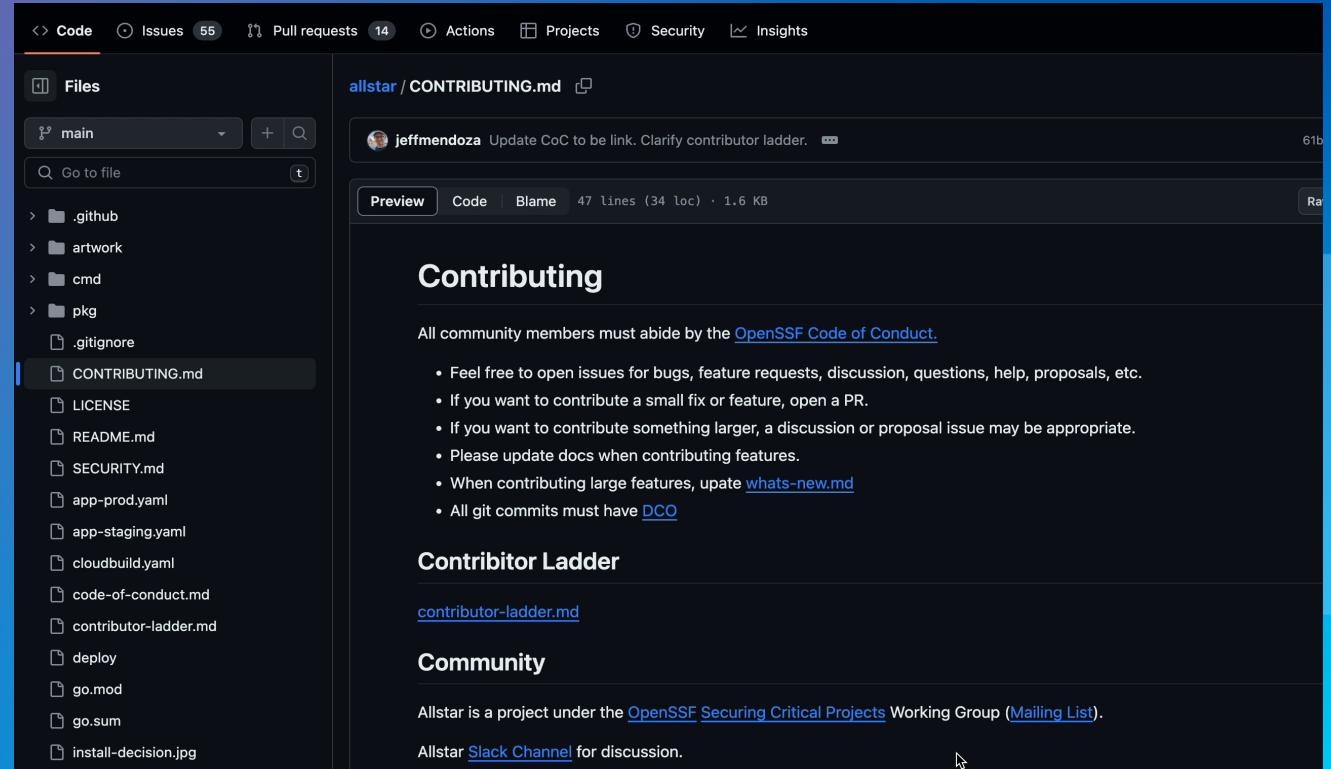
### 3. Evaluating software: Maintenance and release management

- Has there been substantial activity in the last year?
- Look at the release cadence
  - Is it documented?
  - Regularly occurring?
  - Prompt patch releases to address bugs and security issues?
- Does the project communicate announcements regularly?  
Does it have a blog?
- Is the latest release a “-alpha” or “-beta,” or does it indicate that it is not yet production-ready?



# 4. Evaluating software: Community engagement

- Contributor guide?
- Extensively used?
- Is the community working toward security best practices?
  - Automated tests
  - Up-to-date dependencies



[Why Drupal?](#)[Use cases](#)[Resources](#)[Services](#)[Community](#)[Events](#)[Download](#)[Give](#)[Demo](#)

# Community



Early Bird Registration for DrupalCon Portland 2024 is open! Register by 23:59 UTC on 18 March 2024, to get \$100 off your ticket.

[Register now](#)[Contributor guide](#)

## Contribution areas

This guide is always evolving. If you'd like to help improve it, the best starting point is the [Contribute to the Contributor Guide](#) page.

The Drupal project has many areas that you can contribute to improving -- it's not

### Help improve this page

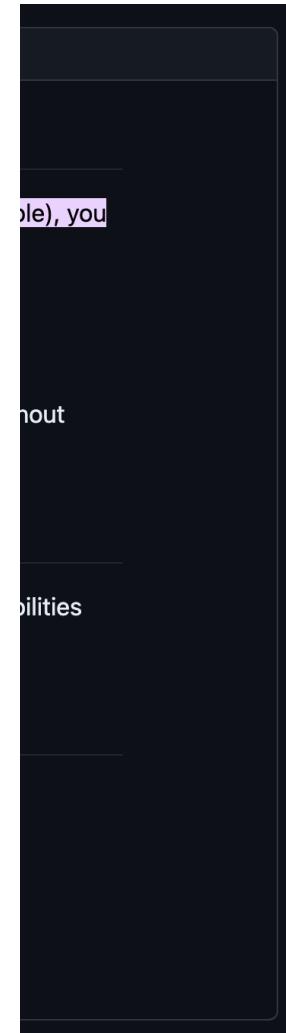
[Create an issue](#) describing the problem.

# 5. Evaluating software: Secure bug reporting

## 🔗 Where do I report security issues?

- If you are here to report any sort of security issue with [a site hosted on WordPress.com](#), then please [submit a report at the Automattic HackerOne page](#). If the issue you're trying to report is on [WordPress.com](#) and is **not** a security issue, then please use their [support forums](#) instead.
- If you're having an issue with your own self-hosted [WordPress.org](#) site that is **not** a security issue, then please use the [WordPress.org support forums](#).
- For security issues with WordPress plugins, follow the information on [Reporting Plugin Security Issues](#).
- For security issues with the self-hosted version of WordPress, submit a report at the [WordPress HackerOne page](#). Include as much detail as you can. Please **always use HackerOne instead of Core Trac**, even if the vulnerability is only in [trunk](#), or a [beta/RC](#) release, because there are some sites that run those in production.

In all cases, you should **not** share the details with anyone else until after the fix for the bug has been officially released to the public.



Tools for securing open source software

# Beyond basic health

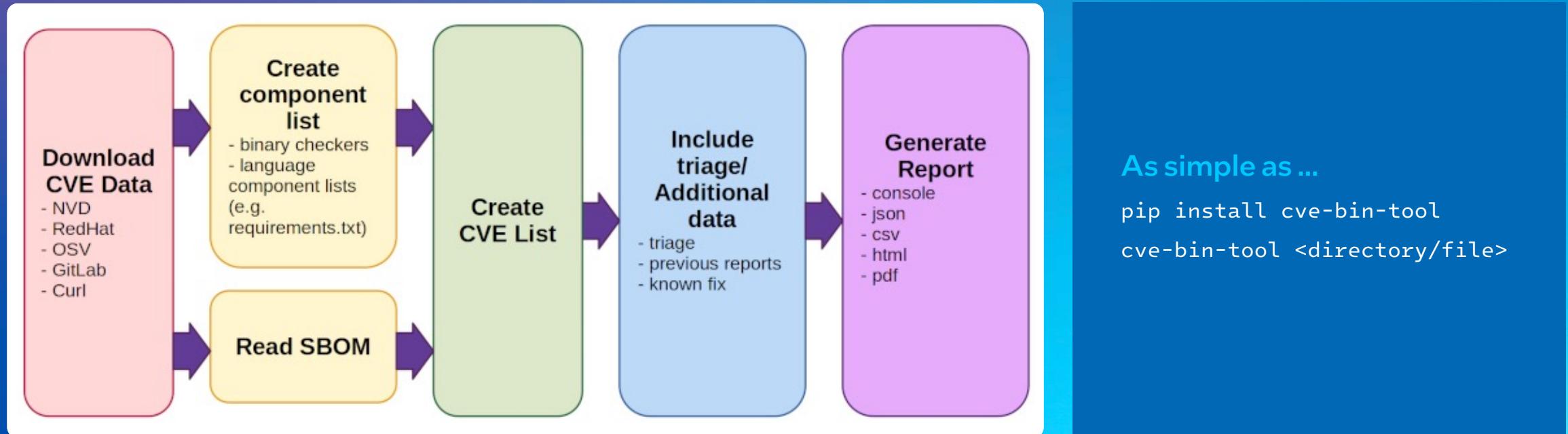
# Security tools for open source software

- Intel maintained CVE-bin-tool
- Open Source Security Foundation (OpenSSF):
  - OpenSSF Best Practices Badge
  - Secure Supply Chain Consumption Framework (S2C2F)
  - OpenSSF Scorecard

# CVE-bin-tool

The CVE Binary Tool can help you find known vulnerabilities in software by using data from the [National Vulnerability Database \(NVD\)](#) list of [Common Vulnerabilities and Exposures \(CVEs\)](#) as well as known vulnerability data from [Redhat](#), [Open Source Vulnerability Database \(OSV\)](#), [Gitlab Advisory Database \(GAD\)](#), and [Curl](#).

1. A binary scanner—Helps you determine which packages may have been included as part of a piece of software.
2. Tools for scanning known component lists—Such as CSV files, SBOM formats, etc.



# Open Source Security Foundation (OpenSSF)

The Open Source Security Foundation (OpenSSF) seeks to make it easy to **develop, maintain, and consume** open source software safely and securely.

This includes fostering collaboration, establishing best practices, and developing innovative solutions for the open source software we all depend on.



# Working groups, projects & SIGs

1. INFORM

## Vulnerability disclosures

*Efficient vulnerability reporting and remediation*

- I. [CVD Guides](#) SIGs
- J. [OSS-SIRT](#) SIG
- K. [Open Source Vuln Schema \(OSV\)](#) project
- L. [OpenVEX](#) SIG
- M. [Vuln Autofix](#) SIG



## Metrics & metadata

*Security metrics/reviews for open source projects*

- N. [Security Insights](#) project
- O. [Security-Metrics: Risk Dashboard](#) project
- P. [Security Reviews](#) project
- AH. [Security Insights Spec](#) project

## Security tooling

*State of the art security tools*

- Q. [SBOM Everywhere](#) SIG
- R. [OSS Fuzzing](#) SIG
- AI. [SBOMit](#) project
- Protobom project



## Securing critical projects

*Identification of critical open source projects*

- U. [List of Critical OS Prj, Components & Frameworks](#) SIG
- V. [Criticality score](#) project
- W. [Harvard study](#) SIG
- X. [Package Analysis](#) project
- Y. [Allstar](#) project



2. EQUIP

## Best practices

*Identification, awareness, and education of security best practices*

- A. [Secure Software Development Fundamentals courses](#) SIG
- B. [Security Knowledge Framework \(SKF\)](#) project
- C. [OpenSSF Best Practices Badge](#) project
- D. [OpenSSF Scorecard](#) project
- E. [Common Requirements Enumeration \(CRE\)](#) project
- F. [Concise & Best Practices Guides](#) SIGs
- G. [Education](#) SIG
- H. [Memory Safety](#) SIG
- AG. [The Security Toolbelt](#) SIG



3. ENGAGE

## End users

*Voice of public & private sector organizations that primarily consume open source*

- Z. [Threat Modeling](#) SIG

## Supply chain integrity

*Ensuring the provenance of open source code*

- S. [Supply-chain Levels for Software Artifacts \(SLSA\)](#) project
- T. [Secure Supply Chain Consumpt Framework \(S2C2F\)](#) project
- AI. [Gittuf](#) project
- AK. [GUAC](#) project



## Securing software repositories

*Collaboration between repository operators*

- AB. [RSTUF](#) project



## Diversity, equity & inclusion

*Increase representation and strengthen the overall effectiveness of the cybersecurity workforce*

## Projects

*Category-leading software initiatives*

- AD. [Alpha-Omega](#)
- AE. [Sigstore](#)
- AF. [Core Toolchain Infrastructure \(CTI\)](#)



# OpenSSF Best Practices Badge

The screenshot shows the OpenSSF Best Practices badge interface for the project "Gramine Library OS with I". The interface includes a sidebar with links like "README", "LGPL-3.0 license", and "Security". The main content area has a heading "Documentation" and four items:

- Documentation**:
  - Met (radio button selected)
  - Unmet
  - N/A
  - ?

The project MUST have a documented roadmap that describes what the project intends to do and not do for at least the next year. (URL required) [\[documentation\\_roadmap\]](#) [Show details](#)
- Architecture**:
  - Met (radio button selected)
  - Unmet
  - N/A
  - ?

The project MUST include documentation of the architecture (aka high-level design) of the software produced by the project. If the project does not produce software, select "not applicable" (N/A). (URL required) [\[documentation\\_architecture\]](#) [Show details](#)
- Security Requirements**:
  - Met (radio button selected)
  - Unmet
  - N/A
  - ?

The project MUST document what the user can and cannot expect in terms of security from the software produced by the project (its "security requirements"). (URL required) [\[documentation\\_security\]](#) [Show details](#)
- Quick Start Guide**:
  - Met (radio button selected)
  - Unmet
  - N/A
  - ?

The project MUST provide a "quick start" guide for new users to help them quickly do something with the software. (URL required) [\[documentation\\_quick\\_start\]](#) [Show details](#)

<https://www.libreoffice.org/get-help/documentation/>

# OpenSSF Scorecard

- What is it?
  - Quick, easy project assessment via list of automated checks for best practices
- What does it help protect me from?
  - Malicious maintainers and packages
  - Poorly maintained projects
  - Compromised build systems and/or code
- How do I use it?
  - Command line interface (CLI)
  - GitHub Action

README Code of conduct Apache-2.0 license Security

## OpenSSF Scorecard

openssf scorecard 9.6 openssf best practices passing build passing CodeQL passing go reference go report A+  
codecov 75% SLSA level 3 slack openssf/scorecard

### Overview

- [What Is Scorecard?](#)
- [Prominent Scorecard Users](#)
- [View a Project's Score](#)
- [Scorecard's Public Data](#)

### Using Scorecard

- [Scorecard GitHub Action](#)
- [Scorecard REST API](#)
- [Scorecard Badges](#)
- [Scorecard Command Line Interface](#)
  - [Prerequisites](#)
  - [Installation](#)
  - [Authentication](#)
  - [Basic Usage](#)



## OpenSSF projects and tools

# OpenSSF Scorecard

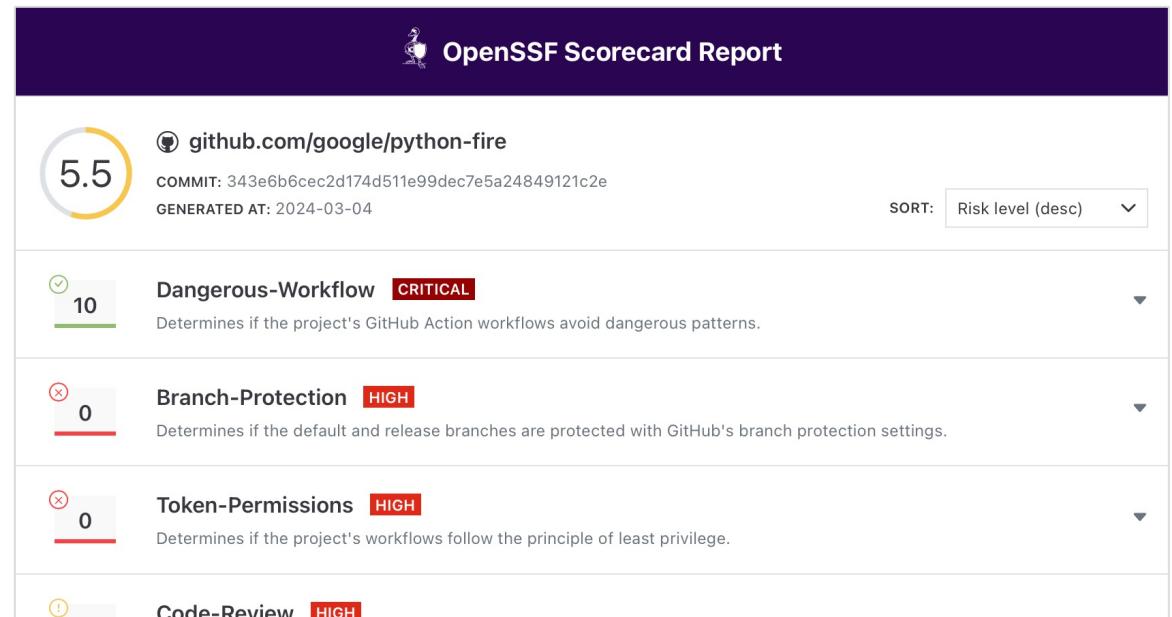
Score in terminal ...

```
Finished [Pinned-Dependencies]
Finished [Fuzzing]
Finished [Packaging]
Finished [Dependency-Libraries]
Finished [Code-Review]
Finished [BAST]
Finished [Dangerous-Workflow]
Finished [License]
Finished [Token-Permissions]
Finished [Dependency-Update-Tool]
Finished [Branch-Protection]
Finished [CI-Tests]
Finished [Maintained]

RESULTS
Aggregate score: 5.1 / 18

Check scores:
SCORE NAME REASON DOCUMENTATION/REMEDIATION
10 / 10 Binary-Artifacts no binaries found in the repo https://github.com/ossf/scorecard/blob/49c8eed3a423f80c872b5c3c9f1bbca9e8aae799/docs/checks.md#binary-artifacts
0 / 10 Branch-Protection branch protection not enabled on default and release branches https://github.com/ossf/scorecard/blob/49c8eed3a423f80c872b5c3c9f1bbca9e8aae799/docs/checks.md#branch-protection
0 / 10 CI-Tests 2 out of 29 merged PRs checked by a CI test -- score normalized to 0 https://github.com/ossf/scorecard/blob/49c8eed3a423f80c872b5c3c9f1bbca9e8aae799/docs/checks.md#ci-tests
0 / 10 CII-Best-Practices no effort to earn an OpenSRE best practices badge detected https://github.com/ossf/scorecard/blob/49c8eed3a423f80c872b5c3c9f1bbca9e8aae799/docs/checks.md#cii-best-practices
6 / 10 Code-Review found 12 unreviewed changesets out of 38 -- score normalized to 6 https://github.com/ossf/scorecard/blob/49c8eed3a423f80c872b5c3c9f1bbca9e8aae799/docs/checks.md#code-review
10 / 10 Contributors 3 different organizations found -- score normalized to 10 https://github.com/ossf/scorecard/blob/49c8eed3a423f80c872b5c3c9f1bbca9e8aae799/docs/checks.md#contributors
10 / 10 Dangerous-Workflow no dangerous workflow patterns detected https://github.com/ossf/scorecard/blob/49c8eed3a423f80c872b5c3c9f1bbca9e8aae799/docs/checks.md#dangerous-workflow
0 / 10 Dependency-Update-Tool no update tool detected https://github.com/ossf/scorecard/blob/49c8eed3a423f80c872b5c3c9f1bbca9e8aae799/docs/checks.md#dependency-update-tool
0 / 10 Fuzzing project is not fuzzed https://github.com/ossf/scorecard/blob/49c8eed3a423f80c872b5c3c9f1bbca9e8aae799/docs/checks.md#fuzzing
9 / 10 License license file detected https://github.com/ossf/scorecard/blob/49c8eed3a423f80c872b5c3c9f1bbca9e8aae799/docs/checks.md#license
10 / 10 Maintained 3 commits out of 38 and 13 issues fixed out of 38 resolved in the last 90 days -- score normalized to 10 https://github.com/ossf/scorecard/blob/49c8eed3a423f80c872b5c3c9f1bbca9e8aae799/docs/checks.md#maintained
? Packaging no published package detected https://github.com/ossf/scorecard/blob/49c8eed3a423f80c872b5c3c9f1bbca9e8aae799/docs/checks.md#packaging
0 / 10 Pinned-Dependencies dependency not pinned by hash detected -- score normalized https://github.com/ossf/scorecard/blob/49c8eed3a423f80c872b5c3c9f1bbca9e8aae799/docs/checks.md#pinned-dependencies
```

... or via browser



# Secure Supply Chain Consumption Framework (S2C2F)

The S2C2F project works to further develop and improve the S2C2F guide, which outlines how to securely consume open source software (OSS) dependencies.

Level 1	Level 2	Level 3	Level 4
 <b>Minimum OSS Governance Program</b> <ul style="list-style-type: none"><li>• Use package managers</li><li>• Local copy of artifact</li><li>• Scan with known vulns</li><li>• Scan for software licenses</li><li>• Inventory OSS</li><li>• Manual OSS updates</li></ul>	 <b>Secure Consumption and Improved MTTR</b> <ul style="list-style-type: none"><li>• Scan for end life</li><li>• Have an incident response plan</li><li>• Auto OSS updates</li><li>• Alert on vulns at PR time</li><li>• Audit that consumption is through the approved ingestion method</li><li>• Validate integrity of OSS</li><li>• Secure package source file configuration</li></ul>	 <b>Malware Defense and Zero-Day Detection</b> <ul style="list-style-type: none"><li>• Deny list capability</li><li>• Clone OSS source</li><li>• Scan for malware</li><li>• Proactive security reviews</li><li>• Enforce OSS provenance</li><li>• Enforce consumption from curated feed</li></ul>	 <b>Advanced Threat Defense</b> <ul style="list-style-type: none"><li>• Validate the SBOMs of OSS consumed</li><li>• Rebuild OSS on trusted infrastructure</li><li>• Digitally sign rebuilt OSS</li><li>• Generate SBOM for rebuilt OSS</li><li>• Digitally sign protected SBOMs</li><li>• Implement fixes</li></ul>

**Putting the tools to work**

# Let's evaluate some software

# Evaluating open source projects

1. Review basic health—Is it active?
2. Check governance—Is it defined?
3. Review maintenance & releases—Is there a cadence?
4. Explore the community—Are people engaged?
5. Bug reporting—Is there a documented process?
6. Run OpenSSF Scorecard

## OpenSSF projects and tools

# Grab a random repo

You can apply some optional filters:

Language

Topic

Next

Save ✓

google/python-fire

Python Fire is a library for automatically generating command line interfaces (CLIs) from absolutely any Python object.

★ 25936 ⚡ 1505 ⚡ 25936 Python

Buy me a coffee



git -random

Picks a random public GitHub repository across all languages and topics. Create a shortlist of repos to view them all at once or save them for later viewing

Download List of Selected Repos Open All

© 2020 DigitalBunker

# Basic health check: Looks promising!

## Python Fire

python 2.7 | 3.5 | 3.6 | 3.7 | 3.8 | 3.9

*Python Fire is a library for automatically generating command line interfaces from Python objects.*

- Python Fire is a simple way to create a CLI in Python. [1]
- Python Fire is a helpful tool for developing and debugging Python code. [2]
- Python Fire helps with exploring existing code or turning other people's code into a CLI. [3]
- Python Fire makes transitioning between Bash and Python easier. [4]
- Python Fire makes using a Python REPL easier by setting up the REPL with the modules and variables you'll need already imported and created. [5]

### Installation

To install Python Fire with pip, run: `pip install fire`

To install Python Fire with conda, run: `conda install fire -c conda-forge`

To install Python Fire from source, first clone the repository and then run: `python setup.py install`

No OpenSSF Best Practices Badge?

Used by 28k



+ 28,004

Contributors 62



+ 48 contributors

Languages



Python 100.0%

# Issue queue

The screenshot shows the GitHub interface for the repository "google/python-fire". The "Issues" tab is selected, showing 122 open issues. A search bar at the top contains the query "is:issue is:open". Below the search bar, there are filters for "Labels" (9) and "Milestones" (0), and a "New issue" button. The main list displays 122 open issues, each with a green circular icon, the issue title, a brief description, and the number of comments (e.g., 3). The issues are listed in chronological order, from oldest at the top to newest at the bottom.

Issue Title	Description	Comments
Cannot parse list of strings containing <code>is</code>	#481 opened 2 weeks ago by rentruewang	3
Remove test requirement on mock	#469 opened on Nov 1, 2023 by dvzrv	3
Version flag alongside other commands	#468 opened on Oct 11, 2023 by amin-nejad	3
[feature request] Exclude function (kw)args from synopsis, arguments and flags in help output	#465 opened on Sep 29, 2023 by eelkeydbos	3
[feature request] support multiple dialects for boolean parameters	#461 opened on Aug 31, 2023 by iRyoka	3
Warning Deprecation: Legacy 'setup.py'	#460 opened on Aug 17, 2023 by nitipit	3
Strings args do not need to be parsed.	#459 opened on Jul 11, 2023 by hxse	3
Fire needs to include features like in ArgParse	#457 opened on Jul 3, 2023 by vihaanmody1	3
[Question] Even if no type hints is supplied, would <code>python-fire</code> accept stub files?	#456 opened on May 16, 2023 by Diogo-Rossi	3
Add Docker Image for easier setup	#455 opened on May 12, 2023 by Faizan-Alam-1	3

# Pull requests

The screenshot shows the GitHub interface for the repository `google/python-fire`. The top navigation bar includes links for Code, Issues (122), Pull requests (26, highlighted), Actions, Projects, Security, and Insights. The repository statistics are shown as Forks (1.5k) and Stars (25.9k). The main content area displays a list of pull requests with the search query `is:open is:pr` applied. The results are filtered by Labels (9) and Milestones (0). A green button for "New pull request" is visible.

PR Number	Title	Author	Comments
#476	Docstring description multiline parsing ✓	thebadcoder96	1 comment, 22 reviews
#467	added venv doc link in readme ✓	krishvsoni	2 comments
#447	#444: Removed pipes dependency ✓	BasedDepartment1	6 comments
#446	Fix pandas.DataFrame support in core._PrintResult ✓	paul-ada	2 comments
#435	ci: watcher for automerge ✘	Borda	20 comments
#432	adding GH dependabot ✘	Borda	17 comments
#426	Support case-insensitive usage	dukecat0	0 comments
#424	Detect the program name when <code>python -m</code> was executed	dukecat0	1 comment, 4 reviews
#408	Support SkipParse decorator ✓	link89	7 comments

# OpenSSF Scorecard: Manual CLI scan—Terminal

```
scorecard --repo github.com/google/python-fire
```



# OpenSSF Scorecard: Manual CLI scan—Browser

 OpenSSF Scorecard Report

**5.5**  [github.com/google/python-fire](https://github.com/google/python-fire)  
COMMIT: 343e6b6cec2d174d511e99dec7e5a24849121c2e  
GENERATED AT: 2024-03-04

SORT: Risk level (desc) ▾

 <b>10</b>	<b>Dangerous-Workflow</b> <span style="background-color: red; color: white; padding: 2px 5px;">CRITICAL</span>	Determines if the project's GitHub Action workflows avoid dangerous patterns.
 <b>0</b>	<b>Branch-Protection</b> <span style="background-color: orange; color: black; padding: 2px 5px;">HIGH</span>	Determines if the default and release branches are protected with GitHub's branch protection settings.
 <b>0</b>	<b>Token-Permissions</b> <span style="background-color: orange; color: black; padding: 2px 5px;">HIGH</span>	Determines if the project's workflows follow the principle of least privilege.
 <b>6</b>	<b>Code-Review</b> <span style="background-color: orange; color: black; padding: 2px 5px;">HIGH</span>	Determines if the project requires human code review before pull requests (aka merge requests) are merged.
 <b>10</b>	<b>Maintained</b> <span style="background-color: red; color: white; padding: 2px 5px;">HIGH</span>	Determines if the project is "actively maintained".
 <b>10</b>	<b>Binary-Artifacts</b> <span style="background-color: red; color: white; padding: 2px 5px;">HIGH</span>	Determines if the project has generated executable (binary) artifacts in the source repository.
 <b>10</b>	<b>Vulnerabilities</b> <span style="background-color: red; color: white; padding: 2px 5px;">HIGH</span>	Determines if the project has open, known unfixed vulnerabilities.

 <b>10</b>	<b>Vulnerabilities</b> <span style="background-color: red; color: white; padding: 2px 5px;">HIGH</span>	Determines if the project has open, known unfixed vulnerabilities.
 <b>0</b>	<b>Fuzzing</b> <span style="background-color: orange; color: black; padding: 2px 5px;">MEDIUM</span>	Determines if the project uses fuzzing.
 <b>0</b>	<b>Pinned-Dependencies</b> <span style="background-color: orange; color: black; padding: 2px 5px;">MEDIUM</span>	Determines if the project has declared and pinned the dependencies of its build process.
 <b>0</b>	<b>SAST</b> <span style="background-color: orange; color: black; padding: 2px 5px;">MEDIUM</span>	Determines if the project uses static code analysis.
 <b>10</b>	<b>Security-Policy</b> <span style="background-color: orange; color: black; padding: 2px 5px;">MEDIUM</span>	Determines if the project has published a security policy.
 <b>0</b>	<b>CII-Best-Practices</b> <span style="background-color: orange; color: black; padding: 2px 5px;">LOW</span>	Determines if the project has an OpenSSF (formerly CII) Best Practices Badge.
 <b>9</b>	<b>License</b> <span style="background-color: orange; color: black; padding: 2px 5px;">LOW</span>	Determines if the project has defined a license.
 ?	<b>Packaging</b> <span style="background-color: orange; color: black; padding: 2px 5px;">MEDIUM</span>	Determines if the project is published as a package that others can easily download, install, easily update, and uninstall.
 ?	<b>Signed-Releases</b> <span style="background-color: red; color: white; padding: 2px 5px;">HIGH</span>	Determines if the project cryptographically signs release artifacts.

# OpenSSF web report: Protocol buffers



## The good:

- No dangerous workflows!
- Maintained!
- Security policy!
- Even fuzzing!



## The less good:

- No signed releases
- Static analysis
- Branch protection unknown

OpenSSF Scorecard Report

github.com/protocolbuffers/protobuf  
COMMIT: 5993a8e996a653d8c6b84d1aeb276bc34a48852e  
GENERATED AT: 2024-02-27T21:44:20Z

SORT: Risk level (desc)

Score	Category	Risk Level	Description
7.5	Dangerous-Workflow	Critical	Determines if the project's GitHub Action workflows avoid dangerous patterns.
10	Signed-Releases	High	Determines if the project cryptographically signs release artifacts.
0	Code-Review	High	Determines if the project requires human code review before pull requests (aka merge requests) are merged.
7	Vulnerabilities	High	Determines if the project has open, known unfixed vulnerabilities.
9	Binary-Artifacts	High	Determines if the project has generated executable (binary) artifacts in the source repository.
10	Dependency-Update-Tool	High	Determines if the project uses a dependency update tool.
10	Maintained	High	Determines if the project is "actively maintained".
10	Token-Permissions	High	Determines if the project's workflows follow the principle of least privilege.
0	Pinned-Dependencies	Medium	Determines if the project has declared and pinned the dependencies of its build process.
0	SAST	Medium	Determines if the project uses static code analysis.
10	Fuzzing	Medium	Determines if the project uses fuzzing.
10	Security-Policy	Medium	Determines if the project has published a security policy.
0	CII-Best-Practices	Low	Determines if the project has an OpenSSF (formerly CII) Best Practices Badge.
9	CI-Tests	Low	Determines if the project runs tests before pull requests are merged.
9	License	Low	Determines if the project has defined a license.
10	Contributors	Low	Determines if the project has a set of contributors from multiple organizations (e.g., companies).
?	Branch-Protection	High	Determines if the default and release branches are protected with GitHub's branch protection settings.
?	Packaging	Medium	Determines if the project is published as a package that others can easily download, install, easily update, and uninstall.

# Get involved in the OpenSSF

Be your own hero!

[openssf.org](https://openSSF.org)

The best way to influence an OSS project direction is to get involved.



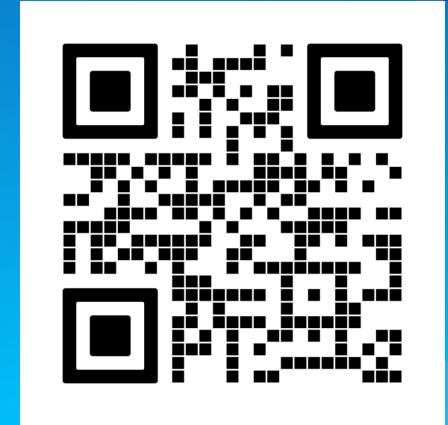
Taking ownership

# Developers don't owe you anything

# Scan for tools, sources, and resources

Visit the presentation's GitHub page

- Intel and OpenSSF tools
- Guides and community resources
- Links to articles and source material
- A PDF of this presentation



# Where to find me

- Twitter/X: @katherined
- Fediverse: @katherined@reality2.social
- LinkedIn: katherinedruckman





#### Notices and disclaimers

Intel is committed to respecting human rights and avoiding complicity in human rights abuses. See Intel's [Global Human Rights Principles](#). Intel® products and software are intended only to be used in applications that do not cause or contribute to a violation of an internationally recognized human right.

Intel® technologies may require enabled hardware, software, or service activation. No product or component can be absolutely secure. Your costs and results may vary.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.